UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/574,064	03/30/2006	Thomas Seidenbecher	2003P11724	9810

24131 7590 09/03/2008

LERNER GREENBERG STEMER LLP
P O BOX 2480
HOLLYWOOD, FL 33022-2480

EXAMINER	
COLUCCI, MICHAEL C	

ART UNIT	PAPER NUMBER
2626	

MAIL DATE	DELIVERY MODE
09/03/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/574,064	SEIDENBECHER, THOMAS	
	Examiner	Art Unit	
	MICHAEL C. COLUCCI	2626	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *13-25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *13-25* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *3-30-06* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

1. Applicants arguments with respect to claims 13-25 have been considered but are moot in view of the new grounds of rejection.

Argument 1 (page 14 paragraph 2):

- "Further, WEBER fails to teach or suggest, among other limitations of Applicant's claims, a text memory containing expressions associated with wildcard character strings that can be present in the dialog egressions of the user interface. In contrast, WEBER contains an array in which an expression originally entered by the user, is stored, after being replaced with a wildcard by the variable replacer of WEBER. As such, the memory array of WEBER cannot be analogized to Applicant's particularly claimed text memory"

Response to argument 1:

Examiner takes Official Notice that it is well known to have a linked relationship between a wildcard/symbol and any variant of a class (i.e. int, string, double type, etc.), wherein during run time of a computer program, strings are substituted for the original variable. However, the string and wildcard relationship must exist at some point in order to substitute for one another. During the compiling of source code, wildcards are substituted in place of strings (double, long, short, etc.), wherein a smaller amount of memory is used corresponding to less address space. This allows for an overall faster processing for the compiler. Additionally, during a user

interaction within a programming operation, rather than using the wildcard, the actual string would be used. For example, if a user defines 'x' to be assigned to the string "enter" and 'y' to be assigned to the string "your name here", during output, the user will not know what “x y” means. Hence, the replacement of symbols x and y during runtime with strings "enter your name here”, allows for clear communication to a user during the transition of compiling to execution. By creating a relationship through the replacement of wildcards and strings, the numbers of entries in a database is reduced, but by the same token, the wildcards will be replaced during the runtime of the program. This can be accomplished through the use of global, date, or run time substitution, wherein there is a direct relationship between a wildcard or symbol and the string. The relationship is handled by a variable manager during the run time of the program code. Without this well known method, strings must be representative of themselves (i.e. name = ‘name’, username = username), which severely limits a users ability as a lexicographer, wherein predefined libraries that contain classes may interfere with a users specific choice of classes to define data within a program. Further, memory reserves would diminish because of excessive variable association, wherein defining a variable to be itself is recursive thereby nullifying the aforementioned method, in which programming itself is based and without substitution of wildcards/symbols, programming can not function as intended.

Examiner takes the position that Weber teaches the aforementioned relationship of wildcard-string pairs, wherein a natural language processor replaces at least one word in the matching phrase prior to searching the database. This may be accomplished by a variable replacer in the natural language processor for substituting a wildcard for the at least one word in the matching phrase. By substituting wildcards for certain words (called "word-variables") in the phrase, the number of entries in the database can be significantly reduced. Additionally, a pronoun substituter in the natural language processor may substitute a proper name for pronouns the matching phrase, allowing user-specific facts to be stored in the database (Col. 3 lines 38-50).

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 13-15, 18-20, rejected under 35 U.S.C. 103(a) as being unpatentable over Weber US 6434524 B1 (hereinafter Weber) in view of Hinks et al. US 5678039 A (hereinafter Hinks) and further in view of Jensen et al. US 6834276 B1 (hereinafter Jenson).

Re claims 13, 19, and 20, Weber teaches selecting a text memory wherein alphanumeric message character strings are associated with alphanumeric identification expressions (Col. 9 lines 42-63);

finding in the text memory identification expressions associated with the wildcard character string contained in the computer program and replacing the wildcard character string in the computer program with the associated message character strings in the text memory (Col. 9 lines 42-63 & Fig. 2), and thereby:

carrying out the finding and replacing steps during a runtime of an executable binary computer program (Col. 9 lines 42-63 & Fig. 2);

carrying out the replacing step by associating the message character strings with memory variables in the running computer programs (Col. 9 lines 42-63 & Fig. 2)

However, Weber fails to teach configuring a language of a computer program (Hinks Col. 3 lines 19-24).

Hinks teaches using editors, which may include a string editor, menu editor, dialog editor, and the like, the end user (translator) can easily access and manipulate the various resources of the program for carrying out translation. The translations themselves are stored back in the Translation Table. Once the end-user translator has completed the task of translating the resources, the translated text is merged back to sources. The Export/Import module is again employed, this time for generating a translated resource file. The translated resource file is similar to the original resource file, except that any necessary translations (e.g., translating an English text string into a French text string) have been carried out.

In addition to translating text strings, other graphical user interface modifications, such as resizing of resources, have also been carried out.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Weber to incorporate replacing text with a wildcard in a computer language as taught by Hinks because editing and replacing text allows for the easy accessing and manipulation of translating resources, where a user can assign wildcards to any function of a natural language or a computer language as a means for customization of short hand notation and keeping track of data type and location (Hinks Col. 3 lines 19-24).

However, Weber in view of Hinks fails to teach checking each dialog element of a user interface of the computer program to determine whether a character string present in the respective dialog element includes a wildcard character string.

Jenson teaches a screen display illustrating an exemplary database registration dialog of a graphic user interface (GUI) embodiment of a database application program implemented according to the present invention on a computer, such as the computer 100. The screen display includes a view options button 500, a database generator button 501, a search button 502, a database display window 504 which provides a list of database names 503, a Register New Database button 505, an UnRegister Selection button 506, and an Enable Word Lists control 507 (Jenson Col. 25 lines 39-56 & Fig. 5).

Further, Jenson teaches a new Link Pattern edit box 1106 can use "wildcard" characters. Wildcard characters are characters such as %, ?, *, and #, where each of the characters has a special meaning. In the embodiment shown, the "%" character substitutes for any digit, the "#" character substitutes for any integer greater than zero, the "*" character substitutes for any number of characters or digits between delimiters, and the "?" character substitutes for any single character (Col. 30 line 66 – Col. 31 line 11).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Weber in view of Hinks to incorporate checking each dialog element of a user interface of the computer program to determine whether a character string present in the respective dialog element includes a wildcard character string as taught by Jenson to allow for a graphical user interface that enables user interaction(Jenson Col. 25 lines 39-56 & Fig. 5), wherein the user can view and alter the database(s) and dialog(s) present within a program, wherein wildcard and string pairs can be identified by a user, wherein in order for a system to compile and execute a program error-free, it must identify a wildcard prior to the replacement of a wildcard/string with a string/wildcard during runtime, the system.

Re claims 14 and 21, Weber teaches the method according to claim 13, which comprises selecting the text memory such that the identification expressions (Col. 9 lines 42-63 & Fig. 2) contain alphanumeric name descriptors

and alphanumeric field descriptors, and a respective field descriptor has an associated message character string (Col. 7 lines 14-30).

Re claims 15 and 22, Weber teaches the method according to claim 14, wherein an identification expression in the text memory is found for a wildcard character string (Col. 9 lines 42-63 & Fig. 2) by evaluating a path (Col. 16 lines 4-30) for the wildcard character string, and wherein the path is formed from at least one of the name descriptors (Col. 7 lines 14-30).

However, Weber fails to teach a character string contained in a computer program (Hinks Col. 3 lines 19-24).

Hinks teaches using editors, which may include a string editor, menu editor, dialog editor, and the like, the end user (translator) can easily access and manipulate the various resources of the program for carrying out translation. The translations themselves are stored back in the Translation Table. Once the end-user translator has completed the task of translating the resources, the translated text is merged back to sources. The Export/Import module is again employed, this time for generating a translated resource file. The translated resource file is similar to the original resource file, except that any necessary translations (e.g., translating an English text string into a French text string) have been carried out. In addition to translating text strings, other graphical user interface modifications, such as resizing of resources, have also been carried out.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Weber to incorporate a

character string contained in a computer program and evaluating a path formed from name descriptors as taught by Hinks because editing and replacing text allows for the easy accessing and manipulation of translating resources, where a user can assign wildcards to any function of a natural language or a computer language as a means for customization of short hand notation and keeping track of data type and location, wherein the use of a path and descriptors is well known, where a path is the particular location of strings in a directory or memory and the use of descriptor in combination with a path evaluation gives a user capability to access files, in order to accomplish a task such as editing (Hinks Col. 3 lines 19-24).

Re claims 18 and 25, Weber teaches the method according to claim 13, which comprises reading the respective wildcard expressions to be replaced from a memory variable in a dialog structure in the computer program (Col. 9 lines 42-63).

4. Claims 16, 17, 23, and 24 rejected under 35 U.S.C. 103(a) as being unpatentable over Weber US 6434524 B1 (hereinafter Weber) in view of Hinks et al. US 5678039 A (hereinafter Hinks) and further in view of Jensen et al. US 6834276 B1 (hereinafter Jenson) and Cseri et al. US 6708164 B1 (hereinafter Cseri).

Re claims 16 and 23, Weber in view of Hinks and Jenson fails to teach the method according to claim 13, which comprising selecting the XML format for

configuring the text memory, and finding the identification expressions by interpreting XML tags (Cseri Col. 8 lines 33-44).

Cseri teaches that a query is constructed to generate a universal table. A universal table includes meta data columns for element tags and meta data columns for parent tags. A universal table also encodes the XML generic identifiers and attribute names in the table column names. Once the element tags and the parent tags are added to a universal table, the universal table fully describes an XML data stream. The present invention is not limited to use with a particular universal table format. Hierarchical results, such as XML data can be produced from a number of universal table formats.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Weber in view of Hinks and Jenson to incorporate an XML table in memory storing tagged expressions as taught by REFB because using XML for the purposes of tagging allows for the editing and labeling or tagging of text and wildcard strings specific to a user, where a user can specify a start or end tag (as well as other tags <title>, <name>, <author>, etc.) relative to a markup language (html, xml, etc.) (Cseri Col. 8 lines 33-44).

Re claim 17 and 24, The Weber in view of Hinks fails to teach the method according to claim 16, which comprises storing identification expressions and message texts in an XML table in the XML text memory (Cseri Col. 8 lines 33-44).

Cseri teaches that a query is constructed to generate a universal table. A universal table includes meta data columns for element tags and meta data columns for parent tags. A universal table also encodes the XML generic identifiers and attribute names in the table column names. Once the element tags and the parent tags are added to a universal table, the universal table fully describes an XML data stream. The present invention is not limited to use with a particular universal table format. Hierarchical results, such as XML data can be produced from a number of universal table formats.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify the system of Weber in view of Hinks and Jenson to incorporate an XML table in memory storing tagged expressions as taught by REFB because using XML for the purposes of tagging allows for the editing and labeling or tagging of text and wildcard strings specific to a user, where a user can specify a start or end tag (as well as other tags <title>, <name>, <author>, etc.) relative to a markup language (html, xml, etc.) (Cseri Col. 8 lines 33-44).

Conclusion

5. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. US 20020105548 A1, US 20030233224 A1, US 7194743 B2.

6. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael C. Colucci whose telephone number is (571)-270-1847. The examiner can normally be reached on 9:30 am - 6:00 pm, Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Richemond Dorvil can be reached on (571)-272-7602. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Michael C Colucci/
Examiner, Art Unit 2626
Patent Examiner
AU 2626
(571)-270-1847
Michael.Colucci@uspto.gov
/Richemond Dorvil/
Supervisory Patent Examiner, Art Unit 2626